線型反復例に対する補外法(数値解析と科学計算)

AUTHOR(S):

長田, 直樹

CITATION:

長田, 直樹. 線型反復例に対する補外法(数値解析と科学計算). 数理解析研究所講究録 1991, 746: 217-229

ISSUE DATE:

1991-03

URL:

http://hdl.handle.net/2433/102209

数理解析研究所講究録
746巻 1991年 217-229

線型反復列に対する補外法

長崎総合科学大学　長田直樹 (Naoki Osada)

1. はじめに

大規模な連立一次方程式

$$Ax = b \tag{1}$$

は，適当な初期値x^0から出発する（SOR法などの）反復法

$$x^{n+1} = Tx^n + c \qquad n = 0, 1, 2, \ldots \tag{2}$$

により計算されることが多い．ここで，Tは正方行列でcは定ベクトルである．Tの固有値は，すべて1ではないものとすると，唯一の固定点

$$x^* = Tx^* + c$$

が存在（$x^* = (I - T)^{-1}c$）する．よく知られている（例えば，[10, p.33]）ように，Tの固有値の絶対値がすべて1未満のとき，反復(2)はx^*に収束する．(2)が発散するとき固定点x^*は，反極限（anti-limit；Shanks[12]の命名）と呼ばれる．

一般に，ベクトル列(x^n)をベクトル列(y^n)に移す変換が，

(y^n)は(x^n)より速く固定点x^*に収束する，

あるいは，

(x^n)は発散するが，(y^n)は固定点x^*に収束する，

を満たすとき，変換 $(x^n) \to (y^n)$ を補外法という．

本報告では，以下の3条件を満たす補外法を考察する：

① (2)によって生成されるベクトル列(x^n)に適用すると理論的には有限回の反復で真値を与える，

② 漸化式により計算でき，逆行列や行列式の計算は不要，

③ ベクトル列の生成方法（例えば，(2)のTとc）については未知でよい．

取り上げる補外法は，Wynnのベクトルε算法（vector ε-algorithm, ＶＥＡ），最小多項式補外（minimal polynomial extrapolation, ＭＰＥ）と修正最小多項式補外（modified minimal polynomial extrapolation, ＭＭＰＥ）である．

本報告の目的は，ＭＰＥとＭＭＰＥをＨ算法で実現し，線型反復列に適用した際の有効性を，数値例によって確かめることである．

2. ベクトルε算法

ベクトルε算法（ＶＥＡ）は1962年にWynnによって提案された算法である[18]．ベクトル列(x^n)に対し，ベクトルε_k^nを

$$\varepsilon_{-1}^n = 0, \quad \varepsilon_0^n = x^n \qquad n = 0, 1, \ldots$$

$$\varepsilon_{k+1}^{n} = \varepsilon_{k-1}^{n+1} + [\varepsilon_{k}^{n+1} - \varepsilon_{k}^{n}]^{-1} \qquad n,k=0,1,\ldots \tag{3}$$

により定義する．ここで，$z\in\mathbb{C}^d$に対し，zの逆ベクトルz^{-1}は$z^{-1}=\bar{z}/\|z\|_2^2$により定義される．

このとき，次の定理が得られている．

定理1（McLeod[9], Graves-Morris[7]）

ベクトル列(x^n)に対し，定数$c_0, c_1, \ldots, c_k$（$c_0, c_k \neq 0$）が存在して，

$$\sum_{j=0}^{k} c_j x^{n+j} = (\sum_{j=0}^{k} c_j) x^* \qquad n=0,1,2,\ldots \tag{4}$$

ならば，

$$\sum_{j=0}^{k} c_j \neq 0 \quad \text{のとき} \quad \varepsilon_{2k}^{n} = x^*, \qquad n=0,1,2,\ldots,$$

$$\sum_{j=0}^{k} c_j = 0 \quad \text{のとき} \quad \varepsilon_{2k}^{n} = 0, \qquad n=0,1,2,\ldots.$$

定理2（Smith, Ford and Sidi[16]）

ベクトル列(x^n)は，反復(2)によって，生成されるものとし，Tのx^1-x^0に関する最小多項式を

$$P(t) = \sum_{j=0}^{k} c_j t^j \tag{5}$$

とする（$P(T)(x^1-x^0)=0$）．このとき，任意の連続するk+1項のベクトル$x^n,\ldots,x^{n+k}$に対し，

$$\sum_{j=0}^{k} c_j x^{n+j} = (\sum_{j=0}^{k} c_j) x^* \tag{6}$$

が成立する．

定理1，2より，(2)によって生成される反復列(x^n)にＶＥＡを適用すると(x^n)が発散する場合でも，（(3)が定義できて，しかも丸め誤差が無ければ）ε_{2k}^{n}が固定点x^*の正確な値を与える．さらに，Tがd次正方行列のとき，$k\leq d$だから，理論的には$x^0,\ldots,x^{2d}$まで計算すると，ＶＥＡによりx^*の値が求められる．

3．ＭＰＥ

ベクトル列(x^n)は，反復

$$x^{n+1} = Tx^n + c \qquad n=0,1,2,\ldots$$

によって，生成されるものとし，Tのx^1-x^0に関する最小多項式を(5)とすると(6)より，

$$\sum_{j=0}^{k} c_j \Delta x^{n+j} = 0 \tag{7}$$

が成立する.

Tの $x^1 - x^0$ に関する最小多項式P(t)が未知のとき, P(t)の係数$c_0, c_1, \ldots, c_k$ ($c_k = 1$) が(7)から求められると, (6)より極限あるいは反極限x^*は

$$x^* = \left(\sum_{j=0}^{k} c_j x^{n+j} \right) / \sum_{j=0}^{k} c_j$$

と表せる.

必ずしも(2)では生成されないベクトル列(x^n)に対し, 最小多項式補外（ＭＰＥ）は, 次のように定義される. (7)を拡張して

$$\left\| \sum_{j=0}^{k-1} c_j \Delta x^{n+j} + \Delta x^{n+k} \right\|_2 \tag{8}$$

を最小にする$c_0, c_1, \ldots, c_k$ ($c_k = 1$) を求め,

$$s_{n,k} = \left(\sum_{j=0}^{k} c_j x^{n+j} \right) / \sum_{j=0}^{k} c_j \tag{9}$$

を補外値にとる. 以上の定式化は, Smith, Ford and Sidi[16]に基づく.

ＭＰＥは, Cabay and Jackson[4]が, (9)を変形した

$$s_{n,k} = \left(\sum_{j=0}^{k-1} \left(\sum_{i=j+1}^{k} c_i \right) \Delta x^{n+j} \right) / \sum_{j=0}^{k} c_j$$

の形で扱ったのが最初である. なお, ＭＰＥの命名はSkelboe[15]による.

ベクトル列(x^n)が(2)により生成されるときは, kを最小多項式の次数とすると(7)の解が存在する（すなわち, (8)の最小値は0になる）ので, (9)は固定点x^*の正確な値を与える. このとき, 理論的には$x^0, x^1, \ldots, x^{k+1}$から真値が得られる.

(8)の最小二乗解は, 正規方程式

$$\begin{cases} c_0 u_{n,n} + \cdots + c_{k-1} u_{n,n+k-1} = -u_{n,n+k} \\ \qquad\qquad \cdots\cdots\cdots \\ c_0 u_{n+k-1,n} + \cdots + c_{k-1} u_{n+k-1,n+k-1} = -u_{n+k-1,n+k} \end{cases} \tag{10}$$

を解くことにより得られる. ここで, $u_{i,j} = (\Delta x^i, \Delta x^j)$である. 正規方程式(10)の解をCramerの公式を用いて表し, (9)に代入して整理すると

$$s_{n,k} = \frac{\begin{vmatrix} x^n & \cdots & x^{n+k} \\ u_{n,n} & \cdots & u_{n,n+k-1} \\ \vdots & & \vdots \\ u_{n+k-1,n} & \cdots & u_{n+k-1,n+k-1} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ u_{n,n} & \cdots & u_{n,n+k-1} \\ \vdots & & \vdots \\ u_{n+k-1,n} & \cdots & u_{n+k-1,n+k-1} \end{vmatrix}} \tag{11}$$

が得られる. ただし, 分子の行列式は第１行について展開したものとする.

(11)を変形した

$$s_{n,k}=\frac{\begin{vmatrix} x^n & \Delta x^n & \cdots & \Delta x^{n+k-2} \\ u_{n,n} & v_{n,n} & \cdots & v_{n,n+k-2} \\ \vdots & \vdots & & \vdots \\ u_{n+k-1,n} & v_{n+k-1,n} & \cdots & v_{n+k-1,n+k-2} \end{vmatrix}}{\begin{vmatrix} v_{n,n} & \cdots & v_{n,n+k-2} \\ \vdots & & \vdots \\ v_{n+k-1,n} & \cdots & v_{n+k-1,n+k-2} \end{vmatrix}}$$

はGermain-Bonne[6]が最初に取り上げたので，Germain-Bonne変換と呼ばれている．ここで，$v_{i,j}=(\Delta x^i, \Delta^2 x^j)$である．

ＭＰＥに対して，nを固定してkのみを動かす場合の漸化式は，Germain-Bonne変換に対し，Brezinski[2]が与えた．nを動かす場合の算法は，最近，Ford and Sidi[5]により得られている．また，５節で述べるＨ算法によっても計算できる．

４．ＭＭＰＥ

Sidi,Ford and Smith[14]は，(7)の代わりに連立１次方程式

$$\sum_{j=0}^{k-1} c_j(y_i, \Delta x^{n+j}) = -(y_i, \Delta x^{n+k}) \qquad i=1,2,\ldots,k \qquad (12)$$

を解くことを提案した．ここで，$y_1,\ldots,y_k$は１次独立なベクトルである．とくに，y_iとして第i成分のみが1でそれ以外が0であるベクトル（単位ベクトル）を取れば，(7)のd個の方程式の最初のk個を解くことになる．(12)から$c_0, c_1, \ldots, c_{k-1}$を求め，(9)により$s_{n,k}$（ＭＰＥと区別するため$t_{n,k}$と表す）を求める方法は，modified ＭＰＥ（ＭＭＰＥ）と名付けられた[14]．このとき，$t_{n,k}$は行列式を用いて

$$t_{n,k}=\frac{\begin{vmatrix} x^n & \cdots & x^{n+k} \\ u_{1,n} & \cdots & u_{1,n+k-1} \\ \vdots & & \vdots \\ u_{k,n} & \cdots & u_{k,n+k-1} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ u_{1,n} & \cdots & u_{1,n+k-1} \\ \vdots & & \vdots \\ u_{k,n} & \cdots & u_{k,n+k-1} \end{vmatrix}} \qquad (13)$$

と表される．ここで，$u_{i,j}=(y_i, \Delta x^j)$である．(13)を最初に提案したのはBrezinski[1]である．

５．Ｈ算法

行列式の比

$$H_k^{(n)} = \begin{vmatrix} x^n & \cdots & x^{n+k} \\ g_1(n) & \cdots & g_1(n+k) \\ \\ g_k(n) & \cdots & g_k(n+k) \end{vmatrix} \Big/ \begin{vmatrix} 1 & \cdots & 1 \\ g_1(n) & \cdots & g_1(n+k) \\ \\ g_k(n) & \cdots & g_k(n+k) \end{vmatrix}$$

を考える．ここで，$(g_i(n))$はスカラ列で，分子の行列式は第1行について展開したものとする．Brezinski[3]は，$H_k^{(n)}$が次の漸化式で表されることを示した．

$n=0,1,2,\ldots$に対し，

$$H_0^{(n)} = x^n, \qquad g_{0,j}^{(n)} = g_j(n) \qquad j=1,2,\ldots,$$

$$H_k^{(n)} = H_{k-1}^{(n)} - g_{k-1,k}^{(n)} \frac{H_{k-1}^{(n+1)} - H_{k-1}^{(n)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}} \qquad k=1,2,\ldots,$$

$$g_{k,j}^{(n)} = g_{k-1,j}^{(n)} - g_{k-1,k}^{(n)} \frac{g_{k-1,j}^{(n+1)} - g_{k-1,j}^{(n)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}} \qquad k=1,2,\ldots : j=k+1,\ldots$$

により定義する．この算法をH算法(H-algorithm)という．名称のHは，Henriciに由来している．

よく知られているように，Henrici[8]は，非線型連立方程式の補外法として以下の方法を提案した：

d次のベクトル列(x^n)に対し，

$$X^n = (x^n, x^{n+1}, \ldots, x^{n+d-1}),$$

$$\Delta X^n = X^{n+1} - X^n, \quad \Delta^2 X^n = \Delta X^{n+1} - \Delta X^n$$

とし，$\Delta^2 X^n$が正則のとき，

$$h^n = x^n - \Delta X^n (\Delta^2 X^n)^{-1} \Delta x^n$$

とおく．ベクトル列(x^n)が，反復(2)によって生成されているとき，ΔX^nと$\Delta^2 X^n$が正則ならば，

$$h^n = x^*.$$

が成立する．Sadok[11]は，H算法において

$$g_j(n) = \Delta x^n{}_j \quad (\text{ベクトル } \Delta x^n \text{ の第j成分}) \qquad j=1,\ldots,d$$

とおくと，

$$h^n = H_d^{(n)}$$

となることを示した．この算法をH-H A (Henrici H-algorithm)という．

MMPEは，H算法において，

$$g_j(n) = (y_j, \Delta x^n) \qquad j=1,\ldots,d$$

とおくことにより,

$$H_k^{(n)} = t_{n,k}$$

が得られる. 反復行列Tの最小多項式の次数kは, 通常は未知であるので,

$$|t_{n,k} - t_{n-1,k}| < \varepsilon \quad \text{または,} \quad |t_{n,k} - t_{n,k-1}| < \varepsilon$$

が成立したとき, 反復を停止し$t_{n,k}$を補外値に採用する. ここで, εは許容誤差である. ＨＨＡはＭＭＰＥの特別な場合であることも注意しておく.

nを固定した場合のＭＰＥもＨ算法により計算できる:

$$H_0^{(j)} = x^{n+j} \qquad j = 0, 1, 2, \ldots$$

$$g_i(n+j) = (\Delta x^{n+i-1}, \Delta x^{n+j}) \qquad i = 1, 2, \ldots : j = 0, 1, 2, \ldots$$

とおくと,

$$H_k^{(0)} = s_{n,k}$$

が得られる. Ｈとgの次元を１つ大きくして

$$H_{n,0}^{(j)} = x^{n+j}, \qquad g_{n,i}(n+j) = (\Delta x^{n+i-1}, \Delta x^{n+j})$$

とすれば, $s_{n,k}$が任意のnに対し計算できる. 次節のＭＰＥは, この方法により計算した. しかしながら, 多くの記憶容量を必要とするので, 実用には適さない.

６. 数値例

6.1連立１次方程式の反復解法

連立１次方程式 $Ax = b$ の解が$x^* = (1, 1, \ldots, 1)^T$となるように, bを決める. 初期値$x^0 = 0$とし,

(a) Jacobi法, (b) Gauss-Seidel法, (c) ＳＯＲ法

により, 反復列(2)を作り,

(i) ＭＰＥ, (ii) ＭＭＰＥ, (iii) ＶＥＡ

により補外（加速）を行う. 比較のため, 加速法として

(v) ＡＡ（田辺[17]の適応的加速法）

も取り上げる. ＡＡはＭＰＥ, ＭＭＰＥ, ＶＥＡとは異なり, 加速の際, 係数行列Aを用いる. (a)(b)(c)および(v)については, [10]を見よ. さらに, 精度と計算量の比較のため, Gaussの消去法による結果も含めた.

数値計算は, 表２の右２列を除いて長崎総合科学大学情報科学センターのACOS 610を使用し, FORTRAN77により倍精度で計算した. 表２の右側２列は, 計算流体力学研究所のS820/80を使用した.

例１　ブロック三重対角行列

$$A=\begin{bmatrix} B & -I & & & & 0 \\ -I & B & -I & & & \\ & -I & B & & & \\ & & & B & -I & \\ & & & -I & B & -I \\ 0 & & & & -I & B \end{bmatrix},\quad B=\begin{bmatrix} 4 & -1 & & & & 0 \\ -1 & 4 & -1 & & & \\ & -1 & 4 & & & \\ & & & 4 & -1 & \\ & & & -1 & 4 & -1 \\ 0 & & & & -1 & 4 \end{bmatrix}$$

Iはm次単位行列，Aはd＝m^2次正方行列である．ＳＯＲ法における加速係数としては，$\omega = 2/(1+\sin(\pi/(m+1)))$ を用いる．

例２　三重対角行列

$$A=\begin{bmatrix} 2 & -1 & & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & & & \\ & & & 2 & -1 & \\ & & & -1 & 2 & -1 \\ 0 & & & & -1 & 2 \end{bmatrix}$$

ＳＯＲ法における加速係数としては，$\omega = 2/(1+\sin(\pi/(d+1)))$ を用いる．

例３　(Wynn[18], Sidi,Ford and Smith[14])

$$A=\begin{bmatrix} 2 & 1 & 3 & 4 \\ 1 & -3 & 1 & 5 \\ 3 & 1 & 6 & -2 \\ 4 & 5 & -2 & -1 \end{bmatrix}$$

Gauss-Seidel法の反復行列

$$T=\frac{1}{18}\begin{bmatrix} 0 & 9 & 27 & 36 \\ 0 & -3 & -3 & 18 \\ 0 & 5 & 14 & 21 \\ 0 & -61 & -151 & -96 \end{bmatrix}$$

の固有値は，$-2.35\pm 2.05i$, -0.0228, と0である[14]．反復(2)は発散するので，ＳＯＲ法は適用しない．

d＝4の場合を表１，d＝16の場合を表２に示す．許容誤差は10^{-9}，最大反復回数は200である．桁あふれ防止のため，中間結果が30桁を越えたら反復を停止する．反復回数n，ＭＰＥ，ＭＭＰＥ，ＶＥＡについてはk（$s_{n-k-1,k}$, $t_{n-k-1,k}$, ε_{2k}^{n-2k}），有効桁数とCPU時間（単位ミリ秒）を示す．×印は，発散したか，目標精度が達成できなかったことを示す．d＝100の場合を表２の右２列に示す．最大反復回数は1000回である．表３-５は，表１と同じく最大反復回数は200である．

表1　連立1次方程式　例1-3（d＝4）

		例　1 n (k) S.D. CPU time	例　2 n (k) S.D. CPU time	例　3 n (k) S.D. CPU time
Jacobi	base	30　9.03 17 m.s.	99　9.04 52 m.s.	×86　-30.09 46 m.s.
	MPE	2(1)　exact 5 m.s.	4(3)　exact 9 m.s.	5(4)　13.60 18 m.s.
	MMPE	3(1)　exact 5 m.s.	4(3)　exact 7 m.s.	6(4)　14.22 11 m.s.
	VEA	2(1)　exact 5 m.s.	5(2)　exact 10 m.s.	8(4)　13.96 19 m.s.
	AA	1　exact 3 m.s.	12　9.61 12 m.s.	×81　-30.19 77 m.s.
Gauss・Seidel	base	16　9.16 10 m.s.	50　9.02 27 m.s.	×59　-30.00 32 m.s.
	MPE	4(3)　16.86 11 m.s.	5(4)　15.65 17 m.s.	5(4)　9.25 17 m.s.
	MMPE	5(1)　exact 9 m.s.	5(4)　15.18 9 m.s.	5(4)　14.04 10 m.s.
	VEA	5(1)　exact 9 m.s.	7(2)　16.56 15 m.s.	7(3)　12.76 15 m.s.
	AA	3　15.65 4 m.s.	9　9.77 9 m.s.	×128　-30.07 119 m.s.
SOR	base ω	10　9.32 7 m.s. 1.0718	19　9.24 11 m.s. 1.2596	
	MPE	4(3)　15.20 11 m.s.	5(4)　15.30 18 m.s.	
	MMPE	5(4)　15.09 10 m.s.	5(4)　14.95 9 m.s.	
	VEA	6(3)　15.24 13 m.s.	8(4)　14.98 19 m.s.	
	AA	6　11.29 7 m.s.	9　9.19 10 m.s.	
Gaussian elimination		exact 2 m.s.	exact 2 m.s.	exact 3 m.s.

表の見方　例1の方程式をGauss-Seidel法で解き，ＭＰＥで補外した結果は，

4(3)　16.86 11 m.s.

例1の7段目

である．$n=4, k=3$のとき$s_{n-k-1,k}=s_{0,3}$が，補外値を与え，

$$16.86=-\log_{10}\|s_{0,3}-x^*\|_\infty, \qquad x^*=(1,1,1,1)^T$$

となり，CPU時間は11ミリ秒（＝0.011秒）である．

表２　連立１次方程式　例１－２

		d＝16 ACOS610		d＝100 S820/80	
		例　1	例　2	例　1	例　2
		n (k) S.D. CPU time	n (k) S.D. CPU time	n (k) S.D. CPU time	n (k) S.D. CPU time
Jacobi	base	100 9.07 379 m.s.	× 200 1.39 752 m.s.	512 9.02 105 m.s.	× 1000 0.11 205 m.s.
	MPE	4(3) 16.16 41 m.s.	10(9) 14.44 140 m.s.		
	MMPE	20(2) 9.74 192 m.s.	21(8) 11.23 239 m.s.	35(19) 9.15 29 m.s.	740(87) 9.11 1899 m.s.
	VEA	6(3) 15.63 47 m.s.	17(8) 11.35 213 m.s.	28(14) 10.36 33 m.s.	× 1000 6.95 11808 m.s.
	AA	28 9.30 215 m.s.	× 200 2.98 1531 m.s.	233 9.01 93 m.s.	× 1000 0.32 399 m.s.
Gauss・Seidel	base	51 9.08 195 m.s.	× 200 2.87 753 m.s.	258 9.04 52 m.s.	× 1000 0.32 202 m.s.
	MPE	9(8) 12.02 254 m.s.	18(16) 9.54 2882 m.s.		
	MMPE	9(4) 15.65 93 m.s.	17(16) 13.18 245 m.s.	33(32) 9.60 32 m.s.	872(*) 9.23 2531 m.s.
	VEA	13(4) 14.63 143 m.s.	28(10) 9.33 496 m.s.	39(11) 9.06 59 m.s.	× 1000 5.58 11798 m.s.
	AA	10 9.80 77 m.s.	38 9.06 290 m.s.	31 9.04 12 m.s.	× 1000 5.74 397 m.s.
SOR	base ω	20 9.42 81 m.s. 1.2596	69 9.12 273 m.s. 1.6895	46 9.16 10 m.s. 1.5604	410 9.01 86 m.s. 1.9397
	MPE	10(9) 14.44 176 m.s.	17(16) 9.24 2362 m.s.		
	MMPE	14(13) 14.70 187 m.s.	18(16) 13.93 272 m.s.	41(40) 10.13 47 m.s.	103(*) 9.73 210 m.s.
	VEA	16(8) 9.85 200 m.s.	33(16) 14.03 666 m.s.	39(19) 9.15 59 m.s.	200(*) 12.56 1347 m.s.
	AA	10 9.22 81 m.s.	35 9.08 280 m.s.	22 9.01 9 m.s.	260 9.01 109 m.s.
Gaussian elimination		15.27 122 m.s.	exact 35 m.s.	13.98 3 m.s.	12.97 3 m.s.

(*)　k＝100

d＝100の場合のＭＰＥは，配列が大きくなりすぎるため，計算できなかった．

6.2線型反復列の加速

線型反復

$$x^{n+1} = Tx^n + c \qquad n = 0, 1, 2, \ldots$$

の例．すべて，$x^0 = 0$ とする．

例４　巡回行列

$$T = \begin{bmatrix} 0 & & & & 0 & \lambda \\ \lambda & 0 & & & & 0 \\ & 0 & \lambda & 0 & & \\ & & & & & \\ & & & \lambda & 0 & 0 \\ 0 & & & & \lambda & 0 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

表３に $\lambda = 0.5$ と 0.9，$d = 4$ と d=16の場合の反復回数と有効桁数を示す．

例５　Jordan行列

$$T = \begin{bmatrix} \lambda & 1 & & & & 0 \\ & \lambda & 1 & & & \\ & & \lambda & & & \\ & & & & & \\ & & & \lambda & 1 & \\ 0 & & & & \lambda & 1 \\ & & & & & \lambda \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

表４に $(\lambda, d) = (0.5, 4)$，$(0.9, 4)$，$(0.5, 16)$ の場合の結果を示す．

例６　(Sidi, Ford and Smith[14])

$$T = 0.06 \times \begin{bmatrix} 5 & 2 & 1 & 1 & & & & & & & 0 \\ 2 & 6 & 3 & 1 & 1 & & & & & & \\ 1 & 3 & 6 & 3 & 1 & 1 & & & & & \\ 1 & 1 & 3 & 6 & 3 & 1 & 1 & & & & \\ & 1 & 1 & 3 & 6 & 3 & 1 & 1 & & & \\ & & 1 & 1 & 3 & 6 & 3 & 1 & 1 & & \\ & & & 1 & 1 & 3 & 6 & 3 & 1 & 1 & \\ & & & & 1 & 1 & 3 & 6 & 3 & 1 & \\ & & & & & 1 & 1 & 3 & 6 & 3 & \\ & & & & & & 1 & 1 & 3 & 6 & 2 \\ 0 & & & & & & & 1 & 1 & 2 & 5 \end{bmatrix}$$

固定点が $x^* = (1, 1, \ldots, 1)^T$ となるように，cを決める．Tの固有値はすべて0と1の間の実数で，$\lambda_1 = 0.8965$，$\lambda_2 = 0.7318, \ldots, \lambda_{11} = 0.0313$ [14]である．反復回数と有効桁数は表５に示す．

例７　(Sidi, Ford and Smith[14])

$$T = 0.04 \times \begin{bmatrix} 12 & 11 & & & & & & & & & 0 \\ 11 & 11 & 10 & & & & & & & & \\ 10 & 10 & 10 & 9 & & & & & & & \\ 9 & 9 & 9 & 9 & 8 & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & & & & & \\ 3 & 3 & 3 & 3 & 3 & \cdot & \cdot & \cdot & 2 & & \\ 2 & 2 & 2 & 2 & 2 & \cdot & \cdot & \cdot & 2 & 1 & \\ 1 & 1 & 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & 1 & \end{bmatrix}$$

固定点が $x^* = (1, 1, \ldots, 1)^T$ となるように，cを決める．Tの固有値はすべて実数

で，$\lambda_1 = 1.2892$，$\lambda_2 = 0.8080, \ldots, \lambda_{12} = 0.0012$ [14]である．反復回数と有効桁数は表5に示す．

表3　線型反復列　例4

	d＝4 λ＝0.5 n (k) S.D.	d＝4 λ＝0.9 n (k) S.D.	d＝16 λ＝0.5 n (k) S.D.	d＝16 λ＝0.9 n (k) S.D.
base	31 9.03	×200 8.15	31 9.03	×200 8.15
MPE	2(1) exact	2(1) 14.09	2(1) exact	2(1) 14.09
MMPE	3(1) exact	3(2) 14.09	3(1) exact	3(2) 14.09
VEA	2(1) exact	3(1) 13.87	2(1) exact	3(1) 14.88

表4　線型反復列　例5

	d＝4 λ＝0.5 n (k) S.D.	d＝4 λ＝0.9 n (k) S.D.	d＝16 λ＝0.5 n (k) S.D.
base	49 9.24	×200 1.83	106 9.10
MPE	6(4) 10.36	×200 5.92	79(12) 9.02
MMPE	6(4) 13.97	* 6(4) 8.70	61(16) 9.44
VEA	8(4) 12.48	*156(4) 6.08	80(16) 9.32

* 許容誤差＝10^{-6}（10^{-9}で計算して達成できるが，計算精度を要求しても．）

表5　線型反復列　例6－7

	例 6 n (k) S.D.	例 7 n (k) S.D.
base	192 9.01	×200 -22.48
MPE	7(6) 10.00	*17(12) 6.30
MMPE	7(6) 14.14	13(12) 12.11
VEA	12(6) 12.50	21(7) 9.89

* 許容誤差＝10^{-6}

7．結論

ＭＰＥ，ＭＭＰＥ，ＶＥＡは線型反復列に対し，反復回数を減少させるという意味で，補外効果があるといえる．とくに，例3や例7のように，もとのベクトル列が発散する場合にも補外列は固定点に収束するので，従来の反復法では解けなかった問題への，反復法の適用が可能となる．計算時間（CPU time）の短縮が伴わない場合もあるが，ＭＰＥについては算法とプログラムを改良することにより，大幅な時間短縮が可能と思われる．大規模な連立1次方程式への適用の実用化は今後の課題である．

謝辞　スーパーコンピュータによる計算を実行し，プログラムの改良についても有益な助言をいただいた，計算流体力学研究所の藤野清次主任研究員に深く感謝します．適応的加速法についての多く文献を提供していただいた，統計数理研究

所の田辺国士教授と，例4および例5を教示いただいた名古屋大学の杉浦洋さんにも感謝いたします.

参考文献

[1] C.Brezinski, Generalisations de la transformation de Shanks, de la table de Padé et de l'ε-algorithme, Calcolo 12(1975), 317-360.

[2] C.Brezinski, Recursive interpolation, extrapolation and projection, J.Comp.Appl.Math. 9(1983), 369-376.

[3] C.Brezinski, About Henrici's method for non linear equations, Symposium on Numerical Analysis and Computational Complex Analysis, Zurich, August 15-17, 1983, unpublished.

[4] S.Cabay and L.W.Jackson, A polynomial extrapolation methods for finding limits and antilimits of vector sequences, SIAM J.Numer. Anal.13(1976),734-752.

[5] W.F.Ford and A.Sidi, Recursive algorithms for vector extrapolation methods, Appl.Numer.Math.4(1988), 477-489.

[6] B.Germain-Bonne, Estimation de la limite de suites et formalisation de procédés d'accélération de convergence, Thèse, Université de Lille I, 1978.

[7] P.R.Graves-Morris, Vector valued rational interpolants I, Numer. Math. 42(1983), 331-348.

[8] P.Henrici, Elements of Numerical Analysis, John Wiley & Sons, N.Y. 1964 （一松信他訳，数値解析の基礎，培風館，1973）

[9] J.B.Mcleod, A note on the ε-algorithm, Computing 7(1971), 17-24.

[10] 森正武，数値解析，共立出版，1973

[11] H.Sadok, About Henrici's transformation for accelerating vector sequences, J.Comp.Appl.Math.29(1990),101-110.

[12] D.Shanks, Non-linear transformations of divergent and slowly convergent sequences, J.Math. and Phys. 34(1955), 1-42.

[13] A.Sidi and J.Bridger, Convergence and stability analyses for some vector extrapolation methods in the presence of defective iteration matrices, J.Comp.Appl.Math. 22(1988), 35-61.

[14] A.Sidi, W.F.Ford and D.A.Smith, Acceleration of convergence of vector sequences, SIAM J.Numer.Anal.23(1986), 178-196.

[15] S.Skelboe, Computation of the periodic stedy-state response of

nonlinear networks by extrapolation methods, IEEE Trans.Circuits and Systems, csa-27(1980),161-175.

[16] D.A.Smith, W.F.Ford and A.Sidi, Extrapolation methods for vector sequences, SIAM Rev.29(1987), 199-233.

[17] K.Tanabe, An adaptive acelerastion of general linear iterative processes for solving a system of linear equations, Proc. 5th. Hawaii International Conference on System Sciences, Honolulu, 1972.
（数理解析研究所講究録129(1971),100-118）

[18] P.Wynn, Acceleration techniques for iterated vector and matrix problems, Math.Comp.16(1962),301-322.